AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS:

1-21. (canceled)

22. (previously presented) A method of verifying the integrity of a software application which is executable in a host terminal, comprising the steps of:

i) using the memory context of the authentic software application during the course of execution for determining at least one series of control instructions forming an executable certificate for the software application, which can be executed by said host terminal during the execution of the software application to be verified;

ii) on the host terminal, executing the software application to be verified, receiving the executable certificate thus determined during step i), and executing the series of control instructions of said executable certificate using the memory context of the software application to be verified;

iii) positively comparing the result thus obtained through the execution of the control instructions in the memory context of said host terminal with the result expected from an authentic software application; and

iv) in the event of a positive comparison, continuing with the execution of the software application to be verified (1, 11).

23. (previously presented) The method according to claim 22, in which the host terminal is provided with a processor, wherein said series of control instructions forming the certificate being coded in a language which can be interpreted by said processor of the host terminal.

24. (previously presented) The method according to claim 22, in which the host terminal is provided with a virtual machine which is capable of emulating a processor, wherein said series of control instructions forming the certificate being coded in a language which can be interpreted by the virtual machine of the host terminal.

25. (previously presented) The method according to claim 22, in which the executable certificate includes a portion of the processing necessary for the satisfactory operation of the authentic application.

26. (previously presented) The method according to claim 22, wherein step i) comprising the steps of, establishing in a secure environment, a map with the memory context of the authentic software application during the course of execution, and determining, from the values of this memory map, the series of control instructions intended to form the executable certificate.

27. (previously presented) The method according to claim 22, wherein, in step ii), the executable certificate for the host terminal emanates from an electronic processing circuit which is physically separated from the host terminal.

28. (previously presented) The method according to claim 22, wherein, in step ii), the recovery of the execution values of the memory context comprising the step of reading the values at the addresses of the various portions of the memory of the host terminal, these portions containing the executable instructions and the data intrinsic to the application to be verified.

29. (previously presented) The method according to claim 22, wherein in step iii), the result obtained by the execution of said series of control instructions produces a signature for the application to be verified, said step iii) comprising the step of calculating this signature by said series of control instructions which uses the values of the memory context of the software application to be verified during the course of execution of the application.

30. (previously presented) The method according to claim 22, wherein the software application comprises instructions which permit said series of control instructions to be loaded and executed in the memory context map by substituting at least one address for executing an instruction of said software application by at least one instruction address of the series of instructions which form the certificate.

31. (previously presented) The method according to claim 22, wherein the series of control instructions is selected in such a manner that the state of the memory context of the software application after the execution of the series of control instructions is identical to the state of the memory context of the software application prior to the execution of the series of control instructions.

32. (currently amended) The method according to claim 22, wherein the series of instructions forming the certificate is transported into a stream of data necessary for the execution of the software application to be verified.

33. (previously presented) The method according to claim 22, wherein the software application to be verified is wholly or partially encoded, the correct deciphering of the software application being achieved in the event of integrity of the software application to be verified.

34. (previously presented) An apparatus for verifying the integrity of a software application which is intended to be executed in a host terminal for accomplishing the method according to claim 22, comprising:

- processing means capable of determining at least one series of control instructions for the software application, which can be executed by said host terminal during the execution of the software application, and which forms an executable certificate of said software application,

- means for moving said executable certificate to the host terminal and executing means for executing the series of instructions forming the certificate on said host terminal during the execution of said software application,

- comparison means for positively comparing the result obtained through the execution of the control instructions with the result expected from an authentic application, and

- means which are capable, in the event of a positive comparison, of

continuing with the execution of the software application to be verified.

35. (previously presented) The apparatus according to claim 34, further comprising a smart card or any other secure circuit which is capable of containing the series of control instructions forming the certificate, the host terminal being provided with a reader for reading a smart card or with a means for communicating with the secure circuit, and the means for executing the software application being provided in order to pick-up, in the smart card or in the secure circuit, the series of instructions forming the certificate during the execution of the software application to be verified.

36. (previously presented) The apparatus according to claim 35, wherein the host terminal comprising means of returning, to the smart card or to the secure circuit, a signature produced by the series of control instructions, and the smart card or the secure circuit additionally comprises a software application verifying means which is capable of validating or invalidating the authenticity of the software application to be verified in dependence on the result of the comparison between a signature produced by the series of control instructions and a value for the signature which is known and previously stored in the smart card or in the secure circuit.

37. (previously presented) The apparatus according to claim 36, wherein the smart card comprises means for modifying the operation of the software application to be verified in the event of a negative comparison.

38. (previously presented) The apparatus according to claim 36, wherein the smart card comprises means for modifying the operation of the software application to be verified in the event of a non-transmission of the signature in conformity with predetermined conditions.

39. (previously presented) The apparatus according to claim 34, further comprising means which are capable of preventing the operation of the software application in the host terminal in the event of a negative comparison.

40. (previously presented) The apparatus according to claim 34, wherein the host terminal belongs to the group formed by data processing apparatuses, digital television decoders, equipment for visualising multimedia contents, micro-computers, smart cards, personal organisers, game consoles, and mobile telephones.

41. (previously presented) The apparatus according to claim 34, wherein the processing means comprising means for determining a plurality of executable certificates which differ from one another according to a selected rate and/or condition.

42. (canceled)

43. (previously presented) The method according to claim 22, further comprising: after step i), a step consisting of inserting the executable certificate into a first stream of data and of processing, through encoding, a second stream of the data necessary for the satisfactory operation of the software application to be verified, prior to said second stream being obtained for processing through the software application to be verified.

44. (previously presented) The apparatus according to claim 35, further comprising means which are capable of inserting the executable certificate into a first stream of data and means for processing, through encoding, a second stream of the data necessary for the satisfactory operation of the software application to be verified, prior to said second stream being obtained for processing through the software application to be verified.

45. (previously presented) The method according to claim 22, wherein the series of control instructions is selected in such a manner that the state of the memory context of the software application after the execution of the series of control instructions is without any modification to the state of the memory context of the software application prior to the execution of the series of control instructions.

46. (previously presented) The apparatus according to claim 34, wherein the processing means comprises means for determining a plurality of executable certificates which differ from one another according to a selected condition.